



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/449,782	11/26/1999	JAMES MCKEETH	MICS:0194	6698
52142	7590	04/24/2009	EXAMINER	
FLETCHER YODER (MICRON TECHNOLOGY, INC.) P.O. BOX 692289 HOUSTON, TX 77269-2289				BROPHY, MATTHEW J
ART UNIT		PAPER NUMBER		
2191				
MAIL DATE		DELIVERY MODE		
04/24/2009		PAPER		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>	
	09/449,782	MCKEETH, JAMES	
	<b>Examiner</b>	<b>Art Unit</b>	
	MATTHEW J. BROPHY	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

1) Responsive to communication(s) filed on 19 January 2009.

2a) This action is **FINAL**.                            2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

4) Claim(s) 1-21 and 23-25 is/are pending in the application.

4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.

5) Claim(s) \_\_\_\_\_ is/are allowed.

6) Claim(s) 1-21 and 23-25 is/are rejected.

7) Claim(s) \_\_\_\_\_ is/are objected to.

8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All    b) Some \* c) None of:

1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

1) <input type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____ .
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)	5) <input type="checkbox"/> Notice of Informal Patent Application
Paper No(s)/Mail Date _____ .	6) <input type="checkbox"/> Other: _____ .

## DETAILED ACTION

1. This office action is in response amendment filed January 19, 2009.

### ***Claim Rejections - 35 USC § 103***

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1, 6, 8, 15 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant's Admitted Prior Art (hereinafter AAPA) in view of "The Windows NT Command Shell" by Tim Hill (hereinafter Hill) (1998).

Regarding Claims 1, 15 and 21, AAPA teaches:

invoking, by an application, a call of a command line utility... wherein the command line utility is a utility executable from a command line prompt (**AAPA Page 1, 17-21 "The conventional technique by which a user application obtains command line utility output is shown in FIG. 1. After a temporary text file is created (block 100), the command line utility whose output is desired is invoked via a standard interface (block 102)."**);

receiving output from the command line utility (**AAPA Page 1, Line 21-22 "Output from the command line utility is piped to the temporary file (block 104),"**);  
storing the command line utility output in a system registry database or shared system memory at a location identified by the identifier (**AAPA Page 1, Line 21-22 "Output**

**from the command line utility is piped to the temporary file (block 104),"** [here examiner interprets "shared system memory" as including a temporary file as discussed in AAPA because the "shared system memory", not explicitly defined in the Applicant's specification (Volatile RAM & Window's Clipboard are given as examples in applicant's specification, while non-volatile memory such as EEPROM & Flash are discussed on Page 7 of applicant's specification), is interpreted to include memory accessed by the Operating system, such as the Disk on which the temporary file of AAPA is created on Pages 1-2 of Applicant's Specification.]; and retrieving, by the application, the command line utility output from the system registry database or shared system memory at the location identified by the identifier. **(AAPA Page 1, Line 22-23 "from which the application extracts and processes the desired data (block 106).")**

AAPA does not explicitly teach:

the application providing an identifier in the call of the command line utility however this limitation is taught by Hill:

**(Hill Page 11, "The > redirection symbol redirects command output to the specified file. For example:**

**C:\>dir >c:\dir.txt**

**This example creates a text file C:\DIR.TXT containing the output of the DIR command. The > symbol can be placed anywhere in the command, but is typically placed at the end of the command. A space is permitted between the > symbol**

**and the file name. If the file specified by the redirection symbol already exists, any existing contents are deleted before the command is executed.”)**

However, it would have been obvious, to one of ordinary skill level in the art, at the time of the invention, to modify the method / system / storage device, as disclosed in AAPA, FIG. 1, using WindowsNT known redirection and piping commands, because piping the output of a script command to a file specified in the call of the command line utility allows the program to specify the exact system memory location (file name) in which the output will be stored. The combination is obvious because teachings are found in the prior art, and combined, with no change in functionality. It is a mere use of common sense by one skilled in the art to select and combine such known elements with no new function, i.e., a predictable result. The predictable result, utility output directly stored to a system storage, at a location identified by an identifier. A subsequently invoked extraction will retrieve the modified values from the temporary file.

**Regarding claim 6, AAPA teaches:**

-providing a system storage (**AAPA Page 1, Line 21-22 “Output from the command line utility is piped to the temporary file (block 104),”**) *[here examiner interprets “shared system memory” and “system storage” as including a temporary file as discussed in AAPA because the “shared system memory”, not explicitly defined in the Applicant’s specification (Volatile RAM & Window’s Clipboard are given as examples in applicant’s specification, while non-volatile memory such as EEPROM & Flash are discussed on Page 7 of applicant’s specification), is interpreted to include memory*

*accessed by the Operating system, such as the Disk on which the temporary file of AAPA is created on Pages 1-2 of Applicant's Specification.] While Hill teaches: ... identifier. (Hill Page 11, “The > redirection symbol redirects command output to the specified file. For example:*

**C:\>dir >c:\dir.txt”**

**Regarding claim 8, Hill teaches:**

-providing an identifier indicating shared system memory. (Hill Page 11, “The > redirection symbol redirects command output to the specified file. For example: C:\>dir >c:\dir.txt”) While AAPA teaches: identifier indicating shared system memory (AAPA Page 1, Line 21-22 “Output from the command line utility is piped to the temporary file (block 104),”) [here examiner interprets “shared system memory” and “system storage” as including a temporary file as discussed in AAPA because the “shared system memory”, not explicitly defined in the Applicant’s specification (Volatile RAM & Window’s Clipboard are given as examples in applicant’s specification, while non-volatile memory such as EEPROM & Flash are discussed on Page 7 of applicant’s specification), is interpreted to include memory accessed by the Operating system, such as the Disk on which the temporary file of AAPA is created on Pages 1-2 of Applicant’s Specification.]

4. Claims 2—5,7, 9-14, 18-20 and 23-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant's Admitted Prior Art (hereinafter AAPA) in view of "The Windows NT Command Shell" by Tim Hill (hereinafter Hill) (1998). in view of USPN 6,182,279 to Buxton and further in view of US Patent 6,681,265 Halva, (hereinafter Hlava.)

Regarding Claim 2, AAPA does not teach:

-providing the identifier comprises providing an identifier that identifies one or more entries in the system registry database.

However, this limitation is taught by Buxton:

(Fig. 2, item 205 and col. 13, lines 14-15, "...registry keys are created..." Also see col. 14, lines 29-59, "To facilitate loading of template onto another system...a number of registration key or subkey are included with template. Each template may have the keys 450A-I, as illustrated in Fig. 4C...Key 450H contains information indicating the name of the storage object in template storage file where initialization data...may be located...Key 450I contains information identifying the CLSID...)

In addition it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the command line redirection (to temporary files) of AAPA with the system register storage of Buxton as the use of the system registry provides the ability to fully use a system registry, avoid dual maintenance in temporary files and avoid inefficient use of programs as recognized by US patent 6,681,265 to

Hlvana (Col 2 Ln 37-46, “Advantageously, this method allows command files to access a data store such as the Windows Registry without some of the problems associated with prior approaches. First, access to the data store can be achieved through a command file which is easier to write and maintain than a program. Secondly, there is no concern about synchronization between the data store and permanent environmental variables used to mimic the data store. Finally, the method allows full use of the data store as intended, that is, as a central repository for configuration type information.”)

**Regarding claim 3, Buxton teaches:**

-providing a root key identifier.

(Col. 11, line 2: “Most OLE object application information is stored in subkeys under the CSLID root key...” Also see col. 17, lines 35-41, “Component loader loads, verifies and checks the license of a component by replacing in registry the InProcessServer 32 entry, i.e. key 450A...and adding additional registry keys 450B-J, as previously described, that will let the component loader (receiving a root key identifier) then load the correct OLE control.”)

**Regarding claim 4, Buxton teaches:**

-providing a sub-key identifier.

(Col. 11, line 2 and col. 14, line 31: To facilitate loading of template...a number of registration or subkey are included with template...")

**Regarding claim 5, Buxton teaches:**

-system registry database comprises an operating system registry database.

(Col. 4, line 49: "Operation of computer system is generally controlled...by operating system software, such as...Windows95...")

**Regarding claim 7, Buxton teaches:**

-providing the identifier comprises providing an identifier indicating the system registry database.

(Col. 10, line 66 – col. 11, line 4: A CLSID identifies the functionality of an object class that can display...access to property values...A subkey is used by an OLE to find out information about the control.")

**Regarding claim 9, Buxton teaches:**

-providing the identifier indicating shared system memory identifies a system clipboard memory.

(Col. 11, line 6: “An FORMATETC...is an OLE data structure which acts in a generalized clipboard format...”)

**Regarding claims 10, Buxton teaches:**

-receiving output directly from the command line output utility.

As an example, a utility modifies (utility output) the registry (col. 8, lines 8-11).

**Regarding claim 11, Buxton teaches:**

-receiving output from the command line output utility through a subsequent command line output routine.

As an example, (col. 8, lines 28-29) “Data items within the registry are retrievable (receive output) via calls (from utility call) to the WIN32 APIs.”

**Regarding claim 12, Buxton teaches:**

-associating each line of command line utility output with a line identifier in the system registry database or shared system memory

As an example, (col. 3, lines 1-9) “Template storage with a means for indexing, including key information associated with the template. “...a memory having one or more locations, means for indexing one or more locations within the memory...” Also col. 13, lines 35-44, templates are stored with an enumerated decimal number: “Each template is stored in an ISTORAGE whose name is unique...and may have the form TEMPLEnnn, where nnn may be a decimal number.”)

**Regarding claim 13, Buxton teaches:**

-setting each line identifier to a value corresponding to a position of that line in the command line utility output.

(Rejection of claim 12 is incorporated and further claim contains limitations as recited in claim 12. Therefore claim 13 is rejected under the same rational as claim 12.)

**Regarding claim 14, Buxton teaches:**

-setting a default value of the provided identifier to equal the total number of command utility output lines stored in the system registry database or shared system memory.

(Rejection of claim 12 is incorporated and further claim contains limitations as recited in claim 12. Therefore claim 14 is rejected under the same rational as claim 12.)

**Regarding claim 18, Buxton teaches:**

-instructions to receive one or more lines of output from the command line utility.

See rejection of limitation in claim 1 above.

-instructions to store each of said one or more lines of output in the system registry database or shared system memory.

As an example, (col. 14, lines 26-29) “The remainder of the operating system registry entries are generated by code (instructions to store) in the template storage DLL and are stored in both registry (store output / modified component data in system storage) and the template.”)

**Regarding claim 19, Buxton teaches:**

-instructions to associate a unique identifier with each of the one or more lines of output stored in the system registry database or shared system memory

See rejection of limitations in claim 2 above.

**Regarding claim 20, Buxton teaches:**

-instructions to set a value associated with the received identifier in the system registry database or shared system memory equal to the number of lines of output stored in the system registry database or shared system memory.

(Rejection of claim 18 is incorporated and further claim contains limitations as recited in claim 12. Therefore claim 20 is rejected under the same rational as claim 12.)

**Regarding claim 23, Buxton teaches:**

-the command line utility comprises a first command line utility, and wherein invoking the call by the application comprises invoking a call to pipe output of a second command line utility to the first command line utility...

-wherein storing the command line utility output comprises storing the command line utility output of the first command line utility.

Col. 8, lines 6-7 disclose the OLE libraries comprise the set of system level services (system utilities). As an example of system utilities (col. 20, lines 17-43) Buxton disclosed reading a sub-key from the registry, use the output to determine the real component CLSID, determine whether a valid certificate and license exist, pipe the relevant character string to a CLSID, etc.

Additionally see rejection of claim 1 above.

**Regarding claim 24, Buxton teaches:**

-the command line utility comprises a first command line utility, and wherein invoking the call by the application comprises invoking a call to pipe output of a second command line utility to the first command line utility...

-wherein storing the command line utility output comprises storing the command line utility output of the first command line utility.

This is a 'program storage device' version of claim 23 above. See rejection of claim limitations in claims 15 and 23 above.

**Regarding claim 25, Buxton teaches:**

-the command line utility comprises a first command line utility, the system further comprising a second command line utility, the application to invoke a call that causes output of the second command line utility to be piped to the first command line utility...

-the location identified by the identifier to store output of the first command line utility.

This is a 'system' version of claim 23 above. See rejection of claim limitations in claims 21 and 23 above.

### ***Response to Arguments***

5. Applicant's arguments filed January 19, 2009 have been fully considered but they are not persuasive.

In remarks, Applicant Argues:

Deficiencies of the Rejection of independent claims 1, 15, and 21

As amended, independent claim 1 recites, inter alia, "storing the command line utility output in a system registry database or shared system memory at a location identified by the identifier." Similarly, amended independent claim 15 recites, inter alia, "instructions for causing the computer to ... store the command line utility output in a system registry database or a shared system memory at a location identified by the identifier. Finally, amended independent claim 21 recites, inter alia, "a system registry database or shared system memory having a location identified by the identifier, the location identified by the identifier to store an output of the command line utility."

In the rejection, the Examiner stated that Hill discloses a "redirection symbol" that "redirects command output to the specified file." Office Action mailed October 20, 2008, page 3. Further, in the rejection, the Examiner stated that "here examiner interprets 'system storage' as including a temporary file as discussed in AAPA because the 'system storage' of the claims is not limited to 'a system registry.'" Id. In contrast, amended claims 1, 15, and 21 each include storing "command line utility" output in "a system registry or shared system memory." Applicant asserts that the "system registry" or "shared system memory" recited in claims 1, 15, and 21 are not temporary files as asserted by the Examiner to be disclosed in AAPA. Further, neither Hill nor AAPA disclose storing "command line utility" output in "a system registry or shared system memory." Thus, claims 1, 15, and 21 clearly distinguish over storing the "command line utility output" in a temporary file, as is asserted to be disclosed in AAPA or Hill.

Examiner's Response:

6. Examiner respectfully disagrees. As described in the rejection above, "shared system memory" as amended in the independent claims is not explicitly defined in the specification. The claims are given their broadest reasonable interpretation in light of

the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). As such, here examiner interprets “shared system memory” as a memory used by the operating system. Therefore, AAPA teaches this limitation because AAPA includes a memory disk to house the temporary file which examiner reads on “shared system memory.”

**In Remarks, Applicant Argues:**

Additionally, Applicant asserts that the cited references, taken alone or in combination, do not disclose “the application providing an identifier in the call of the command line utility” as recited by claims 1, 15, and 21....

In the rejection, the Examiner cited the redirection symbol, “>”, as disclosing an “identifier in the call of the command line utility.” Office Action mailed October 20, 2008, page 3. However, the redirection symbol (“>”) is not a “call of the command line utility.” As clearly stated in Hill, “[c]ommand redirection symbols are not visible to the command.” Hill, page 10. Further, Hill states that “the shell processes them before the command is executed and they are not passed as arguments to the command.” Id. Thus, the redirection symbol (“>”) is not any call of the command line utility. For example, the command line utility does not process the redirection symbol (“>”). Indeed, as stated in Hill, the shell processes (e.g., executes) the redirection facility before the “command,” e.g., command line utility, itself.

Further, this redirection symbol is not an “identifier.” As described in Hill and as known by those having ordinary skill in the art, the > symbol does is a “reserved shell character” that provides redirection of an output from the command line utility. The “>” symbol does not and cannot change, because it is a reserved shell character. Thus, the redirection symbol (“>”) is incapable of identifying anything, as it cannot be altered to identify any particular data. Further, Applicant asserts that the “specified file” after the redirection symbol (“>”), such as “dir.txt” described in Hill, is not an identifier in the call of the command line utility. The “specified file” is an argument of the redirection symbol. For example, as shown in table 2.4 of Hill, the “command redirection symbols” are provided as “>file” wherein “file” is the name of the specified file. Hill, page 10. When using the redirection symbol, the command line utility does not receive the “specified file” as a call. Instead, the “specified file” is a part of and is used by the redirection symbol. Thus, Hill does not disclose “the application providing an identifier in the call of the command line utility” as recited in independent claims 1, 15, and 21.

**Examiner’s Response:**

Examiner respectfully disagrees. First, it should be noted that examiner considers the filename used to redirect command utility output as the “identifier” of applicant’s claims. Therefore, Applicant’s arguments with regards to the “>” symbol are misplaced and moot. Second, regarding Applicant’s arguments concerning the “dir.txt” example of Hill,

while the specified file is an argument for a redirection symbol, it is also provided by the console application in the line calling the DIR command line utility. This anticipates applicant's claim limitation "providing an identifier in the call of the command line utility" because the console application provides the identifier of the output redirection destination in the same line as the call to the command line utility. As a result this rejection is maintained.

In remarks, Applicant Argues:

Claims 2-14 are dependent on claim 1, claims 16-10 are dependent on claim 15, and claims 23-25 are dependent on claim 21. As discussed above with regard to the first ground of rejection under 35 U.S.C. § 103(a), Hill does not disclose storing the "command line utility output" in a "system registry database or shared system memory." Buxton and/or Halva do not cure this deficiency of Hill with regard to the base claims 1, 15, and 21. Accordingly, the cited combination does not disclose or suggest all of the elements of the claimed invention, and thus, cannot possibly render the claimed subject matter obvious.

Further, Applicant asserts that Buxton teaches away from such a combination with Hill and Halva. Hill is a reference directed to the "Windows NT Command Shell." Hill, page 1. The descriptions in Hill concern usage of the "command shell," a "command prompt," i.e., a command line, and various commands executed from the "command shell" by typing these commands into the "command prompt." Id. Similarly, Halva is directed to "command files" that are described therein as "a file containing one or more command line operations." Halva, col. 4, lines 10-20. Thus, both Hill and Halva are directed to usage of the "command line" and various commands executed from the command line.

In contrast, Buxton discloses "OLE libraries" that are defined as "system-level services which utilize the interfaces defined by the COM specification" that call a "WIN 32 API." Buxton, col. 8, lines 6-8. Applicant asserts that there is a clear difference between a service and a command executed from the command prompt as recited in Hill, and between a service and a command line operation as recited in Halva.

Further, as known to those of ordinary skill in the art and as stated in Buxton, API's are "application program interfaces" which are also quite different than a utility and a "command line utility." As they are described in Buxton, neither "application program interfaces" nor "system-level services" are "executable from a command line prompt," and thus cannot be considered a "command line utility." Applicant asserts one skilled in the art would not seek to combine Hill and Halva, directed to command line usage, with Buxton, directed to usage of system-level services, e.g., OLE libraries.

Examiner's Response:

Examiner respectfully disagrees. One of ordinary skill in the art at the time of the invention would be motivated to store command line output such as that in Hill, in a

System Registry because Halva teaches that it would be advantageous to store information in the system registry, as “[t]he Windows.RTM. Registry is increasingly used to store application configuration information... Microsoft's policy [is] making the Registry the repository for application configuration type information.” (Halva Col. 1-2). Additionally, The windows registry provides a centralized data store for computer data and direct storage to it avoids the use of temporary files. it is increasingly used to store information. Buxton teaches an API call to achieve registry storage in the window's registry as shown in FIG. 2, 240, 250. While, as described by the applicant, Buxton teaches OLE libraries a.k.a. "system level services" to access this API, One of ordinary skill in the art would be motivated to replace such programs with the command line operations of Hill, because Halva recognized that using command line utilities to interact with a Win32 API is preferable to programs: “To provide access to Registry data, utility functions can be coded as programs rather than command files. These programs can then access the Registry data by using the Windows APIs. However, this method is somewhat inefficient. Programs are more difficult and time consuming to write and maintain than command files. Therefore, this method detracts from developer productivity relative to a method that uses command files to access the Registry data.”

In Remarks, Applicants Argue:

In rejecting claim 9, the Examiner cited the "FORMATEC" of Buxton, which Buxton describes as an "OLE data structure which acts in a generalized clipboard format." Applicant asserts that this description cannot disclose the "system clipboard memory" of claim 9. First, Applicant notes that the "FORMATEC" of Buxton only has a "generalized clipboard format." That is, the "generalized clipboard format" is merely a description of the format of this data structure, and not a description of how the data structure is used.

Nowhere in Buxton is this "FORMATEC" data structure described as providing a "system clipboard memory." Additionally, as stated above, "OLE libraries" are defined as "system-level services." Buxton, col. 8, lines 6-8. Applicant asserts that these "services" as described in Buxton do not provide a "system clipboard memory." Instead, the "FORMATEC" OLE data structure is "used as a parameter in OLE functions and methods that require data format information." Id., col. 11, lines 9-11. A "parameter" for "OLE functions and methods" is not a "system clipboard memory" as recited in claim 9. Accordingly, Applicant respectfully requests withdrawal of the rejection under 35 U.S.C. § 103(a) and allowance of claim 9.

Examiner's Response:

7. Examiner respectfully disagrees. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies are not recited in the rejected claim(s). Specifically, Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Here, while applicant contends that the claim is not taught, the clipboard format structure taught by Buxton anticipates the limitation "providing the identifier indicating shared system memory identifies a system clipboard memory."

***Conclusion***

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to MATTHEW J. BROPHY whose telephone number is

571-270-1642. The examiner can normally be reached on Monday-Thursday 8:00AM-5:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MJB

4/7/2009

/Wei Y Zhen/

Supervisory Patent Examiner, Art Unit 2191

MJB

4/10/2009